33
7-11-03
DW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Art Unit: 2183

Examiner: Richard Ellis

Serial No.: 09/385,394

Filed: August 30, 1999

Applicant(s): John S. Yates, Jr., et al.

Title: COMPUTER WITH TWO EXECUTION MODES

I certify that this correspondence, along with any documents referred to therein, is being transmitted by facsimile on _____ to The Commissioner for Patents, Washington D.C. 20231.

_____

COMMISSIONER FOR PATENTS
Washington D.C. 20231

DECLARATION OF DAVID R. LEVINE

I, David R. Levine, declare as follows.

1. I hold the following degrees:
   - Ph.D., Computer Science, Stanford University (1973)
   - M.S., Computer Science, Stanford University (1968)
   - B.A., Physics, Harvard University (1965)
2. My professional experience includes:
   - 37 years working as a computer scientist in a variety of areas, including operating systems, programming language design, compilers and optimization, and instruction set architecture design.
   - Assistant Professor of Computer Science at Rutgers University
   - Assistant Professor of Computer Science at Boston College
   - Co-inventor on U.S. Patent No. 6,332,188 for features in the instruction set architecture of Analog Devices' TigerSHARC processor
   - Represented Hewlett-Packard company on the ANSI Fortran Language Standards committee
   - Participated in the design of calling conventions for several computers, including IBM System/360, HP PA-RISC, Analog Devices TigerSHARC and ADSP-219x, and Intel Micro-Signal Architecture
   - Researched and designed translator for assembly language to allow moving programs from one computer architecture to another

3. In preparing this affidavit, I have reviewed the following portions of the prosecution history for application serial no. 09/385,394:

- examiner's paper no. 20
- examiner's paper no. 19
- paragraphs 13 and 17 of examiner's paper no. 8

and U.S. Patent No. 5,481,684 (Richter).

4. To the extent that the following paragraphs state definitions for terms, I am relying on my personal knowledge of the plain meaning given those terms in the art and the interpretation given those terms by those of ordinary skill in the art.

I. Big-Endian vs. Little-Endian Access

5. Richter's big-endian/little-endian mode switch, as described by Richter, controls only the connection between the computer's data registers and memory. Modifying that connection is sufficient to implement the capabilities that Richter describes. Richter makes no mention that his mode switch has any affect on the way that data are stored in registers, nor that it has any effect on the connection between the computer's data registers and the arithmetic or logical units of the computer, nor that it has any effect on the internal operation of the processor. When Richter's computer switches between big-endian and little-endian mode, there is no indication that he intended for the contents of the data registers to be altered. Richter does not discuss any mechanism for byte-reversing the contents of a register after the contents exist in a register. None of these techniques is required to implement the capabilities that Richter describes.

6. If a computer's memory content is stored in big-endian form, and the computer attempts to access that content in little-endian mode (or vice-versa), the access will not give a meaningful result, without some explicit correction. For example, all attempts to move a 4-byte or 8-byte number from memory to a data register in a mismatched access mode will result in incorrect results (except for the rare exception of a palindromic datum). If the little-endian/big-endian access mode used to access a memory location mismatches the convention under which a datum is stored in memory, no further useful computation can be performed on that datum without some explicit correction. Richter does not teach or suggest that any such explicit correction is applied, and no such correction is required to implement the capabilities described by Richter.

7. Paragraph 1 of examiner's paper no. 20 admits that, under Richter's little-endian access, the order of the bytes will be altered as they are moved from memory to a register. Paragraph 1 errs in stating that the value "remains the same" after the byte order is reversed. Changing the order of bytes changes the value represented, just as reversing the order of digits in an everyday base-10 number changes the value.

8. Paragraph 1 of examiner's paper no. 20 asserts that if a 32-bit unsigned integer, stored in memory in big endian order, has a value of 234,567, then a little-endian load into a register of that integer will place the value 234,567 into the register. This assertion of paragraph 1 is incorrect. Rather, a little-endian load of 234,567 (0003A76B in hexadecimal) stored in big-endian form will place the value 113,310,374 (6BA70300 in hexadecimal) into the register. Richter does not discuss any mechanism for converting the value 113,310,374 in the register to 234,567. Richter does not discuss any mechanism for accessing the register in a mode in which the value 113,310,374 in a register will be interpreted as 234,567.

9. Paragraph 1 of paper 20 is incorrect. "Performing a big-endian to little-endian byte-reversal," which is illustrated by the example above, does result in a change in the numerical interpretation of the bits.

10. A computer that behaved as described in paragraph 1 of examiner's paper no. 20 could not reasonably be expected to succeed. Richter's disclosure would not enable one of ordinary skill in the art to develop a computer that operated as described in paragraph 1.

11. Paragraph 2 of examiner's paper no. 20 makes assumptions that are incompatible with the assumptions underlying paragraph 1 of examiner's paper no. 20 and paragraph 3 of examiner's paper no. 19. For example, the latter two paragraphs assume that a single copy of a datum exists in memory, and consider the results of several different kinds of moves from that memory to a register. Paragraph 2 of examiner's paper no. 20 assumes that two separate copies of a stored datum exist, but does not consider the result of any move of any datum. These two analyses are not logically related to each other, and cannot be combined to support any meaningful inference.

12. Paragraph 2 of examiner's paper no. 20 errs in its assertion of "logical equivalence." A computer architect would not consider a memory content of "234,567" to be "logically equivalent" to a register content of "113,310,374."

13. Richter's disclosure is directed to a mechanism that neither involves nor requires mixing little-endian and big-endian access to a particular storage location. Nothing in Richter suggests that Richter even considered it possible to mix meaningful or practical little-endian and big-endian access to the same memory location.

14. The capabilities discussed by the Examiner in paragraphs 1 and 2 of examiner's paper no. 20 and in paragraph 3 of examiner's paper no. 19 are not among the capabilities that Richter ascribes to his computer. The assertions of paragraph 1 of examiner's paper no. 20 attributing certain capabilities to Richter are incorrect. These capabilities are the "invention" of the examiner, not of Richter. Richter does not teach techniques that, without substantial contributions from other art, would achieve the capabilities discussed by the examiner with any reasonable expectation of success.

15. Paragraph 13 of examiner's paper no. 8 discusses "adjusting the data storage content of the computer." No one of ordinary skill in the art would interpret this phrase as embracing a change from big-endian to little-endian access mode, without changing any value of any data storage element of the computer.

II. "Computer Architecture" and "Data Storage Convention"

16. One of ordinary skill in the art would understand that the phrases "computer architecture" and "data storage convention" refer to two different things. A discussion of one does not necessarily imply discussion of the other.

III. "Segment" vs. "Page"

17. The word "segment" has two different interpretations among those of ordinary skill in the art. The two interpretations are quite different from each other. At col. 6, lines 62-67, Richter uses the term "segment" in a manner that makes clear to one of ordinary skill in the art that Richter intends an interpretation for "segment" that relates to memory protection. Richter, along with essentially all other discussions in the art of the Intel X86 architecture, uses the word "segment" to refer to a feature of a memory protection system.

18. In contrast, the definition provided at paragraph 5 (pages 2-3) of examiner's paper no. 19, based on Rosenberg's dictionary, refers to a different definition of "segment" that is irrelevant Richter's use of the word "segment."

19. The art uses the word "page" to refer to a feature primarily related to implementation of a virtual memory system.

20. In view of Richter's express statement of the differences between his "segments" and "pages," at col. 6, lines 62-67, no one of ordinary skill would interpret the word "page" as having any relationship to Richter's "segments," let alone a complete identity.

21. I express no opinion on any issue not expressly set forth above.

22. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Respectfully submitted,

Dated: April 10, 2003

By: David R. Levine